Applicants:	DEVOR, Harold Theodore et al.	Assignee:	Intel Corporation
Serial Number:	10/721,879	Attorney Docket:	P-6216-US

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the Application. Please add or amend the claims to read as follows, and cancel without prejudice or disclaimer to resubmission in a divisional or continuation application claims indicated as cancelled:

1. (Currently Amended) A method comprising:
during translation of a code block from a first format suitable for a first computing platform to a second format suitable for a second computing platform, ~~performing instrumentation of~~ inserting one or more instructions in said code block to detect whether execution of said code block results in the misaligned data access prior to execution of said code block; and
modifying said code block according to said misaligned data access.

2. (Cancelled)

3. (Currently Amended) The method of claim 1 wherein detecting comprises ~~performing instrumentation of~~ inserting at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

4. (Currently Amended) The method of claim 1, wherein ~~performing instrumentation of~~ inserting one or more instructions in said code block comprises ~~performing instrumentation of~~ inserting at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

5. (Original) The method of claim 1, wherein modifying comprises adding to said code block an instruction to branch an execution of said code block to a code sequence whose execution handles the misaligned data access.

6. (Original) The method of claim 1, wherein modifying comprises modifying said code block to handle misaligned data access in a subsequent execution of said code block.

7. (Original) The method of claim 1, further comprising translating said code block from said first format to said second format.

8. (Currently Amended) The method of claim 1, wherein ~~performing instrumentation of~~ inserting one or more instructions in said code block further comprises ~~performing instrumentation of~~ inserting one or more instructions in said code block to detect whether execution of said code block results in the misaligned data access prior to execution of a code block translated from a format suitable for a 32-bit based computing platform to a format suitable for a 64-bit based computing platform.

9. (Currently Amended) An apparatus comprising:
a processor to ~~perform instrumentation~~ insert one or more instructions, during translation of a code block from a first format suitable for a first computing platform to a second format suitable for a second computing platform, ~~of~~ in said code block to detect whether execution of said code block results in misaligned data access prior to execution of said code block, and to modify said code block according to said misaligned data access.

10. (Cancelled)

11. (Currently Amended) The apparatus of claim [[11]] 9 wherein the processor is able to ~~perform instrumentation of~~ insert at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

12. (Currently Amended) The apparatus of claim 9, wherein the processor is able to ~~perform instrumentation of~~ insert at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

13. (Original) The apparatus of claim 9, wherein the processor is able to add to said code block an instruction to branch an execution of said code block to a code sequence whose execution handles the misaligned data access.

14. (Original) The apparatus of claim 9, wherein the processor is able to modify said code block to handle misaligned data access in a subsequent execution of said code block.

15. (Currently Amended) The apparatus of claim 9, wherein the processor is able to, before ~~performing instrumentation~~ insertion, translate said code block from said first format to said second format.

16. (Original) The apparatus of claim 9, wherein the first computing platform is a 32-bit based computing platform and the second computer architecture is a 64-bit based computing platform.

17. (Currently Amended) A computing platform comprising:
a processor to ~~perform instrumentation~~ insert one or more instructions, during translation of a code block from a first format suitable for a first computing

platform to a second format suitable for a second computing platform, ~~of~~ in said code block to detect whether execution of said code block results in misaligned data access prior to execution of said code block, and to modify said code block according to said misaligned data access; and

a dynamic random access memory operably associated with said processor to store at least a portion of said code block.

18. (Cancelled)

19. (Currently Amended) The apparatus of claim 17 wherein the processor is able to ~~perform instrumentation of~~ insert at least one instruction of said code block to detect whether execution of said code block results in the misaligned data access.

20. (Currently Amended) The apparatus of claim 17, wherein the processor is able to ~~perform instrumentation of~~ insert at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

21. (Currently Amended) A machine-readable medium having stored thereon a set of instructions that, if executed by a machine, cause the machine to perform a method comprising:

during translation of a code block from a first format suitable for a first computing platform to a second format suitable for a second computing platform, ~~performing instrumentation of~~ inserting one or more instructions in said code block to detect whether execution of said code block results in the misaligned data access prior to execution of said code block; and

modifying said code block according to said misaligned data access.

22. (Cancelled)

23. (Currently Amended) The machine-readable medium of claim 21 wherein the instructions that result in detecting result in ~~in performing instrumentation~~ insertion of at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

24. (Currently Amended) The machine-readable medium of claim 21, wherein the instructions that result in ~~performing instrumentation~~ insertion result in ~~performing instrumentation~~ insertion of at least one instruction in said code block to detect a location of an instruction whose execution results in the misaligned data access.

25. (Original) The machine-readable medium of claim 21, wherein the instructions comprise at least part of a translator.

26. (Original) The machine-readable medium of claim 21, wherein the instructions comprise at least part of an execution layer.

27. (Original) The machine-readable medium of claim 21, wherein the instructions comprise at least part of an operating system.

28. (Original) The machine-readable medium of claim 21, wherein the instructions comprise at least part of a compiler.